1.0 INTRODUCTION

This plan is the software management plan for the XYZ Project. The XYZ Project is the Acquirer of software capabilities and related services. These capabilities and related services may be provided by other NASA organizations, Universities, general and mission support contractors or contractors providing end-items with embedded and support software. Throughout this document the term "Project" is used to refer to the XYZ Project in its role of software Acquirer.

Throughout this document the term "provider" is used to refer to developers and providers of software and software services regardless of the nature of their organization or their affiliation with the Project.

Each provider of software to the Project shall develop and submit for Project approval a Software Management Plan that follows the organization, format, and content of the Management Plan Data Item Description (NASA-DID-M000) in NASA Software Documentation Standard (NASA-STD-2100-91). It shall respond to each of the provider requirements given in this document.

1.1 Identification of Document

This is the Software Management Plan (SMP) for the Project.

1.2 Scope of Document

This SMP establishes the Project's management processes for the software to be acquired by the Project to satisfy its requirements. It also contains requirements to be satisfied by the providers of all software purchased, contractually acquired, developed or maintained for the support or execution of the Project. Its provisions apply to all Government organizations, in-house activities and contractors providing software capabilities and/or support to the Project. Except for software and hardware interfaces to Project capabilities and schedules for the availability of support resources, it does not apply to operational institutional capabilities that are not specifically developed or modified to support the Project.

The term "software" as used in this document includes code, documentation, associated data, and "firmware", which is software installed in a medium that cannot be dynamically changed.

1.3 Purpose and Objective of Document

The purpose of this SMP is to define software management processes to be followed by the Project, and responsibilities, standards, procedures and organizational relationships for all software activities associated with the Project. It establishes software acquisition and development practices, standards, and

technical procedures. It establishes management, engineering, and assurance requirements for providers of software.

1.4 Document Status and Schedule

This is the initial version of the Plan. It will be reviewed by the software manager at each major system review, or at six month interval, which ever is shorter. Needed changes will be made subsequent to each review.

1.5 Document Organization and Roll-Out

The organization, format, and content of this plan follow the Management Plan Data Item Description (NASA-DID-M000) specified by NASA Software Documentation Standard (NASA-STD-2100-91).

This SMP is organized as follows:

- Section 1 - provides an overview of the context, structure, and content of this Software Management Plan (SMP).
- Section 2 - identifies documents that contain the requirements and references used in this SMP.
- Section 3 - identifies and provides a description of the software that the Project will acquire.
- Section 4 - defines the business practices to be used by the Project and business practice requirements for providers.
- Section 5 - defines the Project's software management practices and sets software engineering requirements for the provider.
- Section 6 - N/A. Development planning shall be included in the provider's Software Management Plan.
- Section 7 - TBS. Describes the Sustaining Engineering and Operations Activities after the software is turned over to the operations staff.
- Section 8 - defines the Project's software assurance program by defining the Project's oversight activities and identifying requirements for the provider.
- Section 9 - identifies the Project's software risk management processes and provider requirements.
- Section 10 - identifies the Project's approach to software configuration management and provider requirements.

- Section 11 - TBS. Describes the Project's processes for delivery and operational transition. Identifies provider requirements for planning for delivery and operational transition activities.

- Section 12 - defines all abbreviations and acronyms used within this document.

- Section 13 - is a glossary of all special terms used within this document.

- Appendix 1 - contains the software documentation requirements imposed on providers.

List of Figures:

2.0 RELATED DOCUMENTATION

2.1 Parent Documentation

This is the top-level document of the Project's Software Documentation Set.

2.2 Applicable Documents

The policies, standards, procedures, and formats contained in the following documents are established as requirements for the Project and all providers.

- IEEE Std 729-1983, IEEE Standard Glossary of Software Engineering Terminology, February 18, 1983.
- NASA-STD-2201-93, Software Assurance Standard, November 10, 1992.
- NASA-STD-2100-91, Software Documentation Standard,
- NASA-STD-2202-93, Software Formal Inspections Standard
- NASA Software Acquisition Life Cycle, Version 4.3
- Software Requirements, Project Document TBS1
- Software Master Schedule, Project Document TBS2
- Software Risk Assessment, Project Document TBS3

2.3 INFORMATION DOCUMENTS

The following documents provide guidance that will assist the providers in complying with Project software requirements identified by this SMP.

- NASA Headquarters Code Q, SMAP-GB-A201, Software Assurance Guidebook, September 1989.
- NASA Headquarters Code Q, SMAP-GB-A301, Software Quality Assurance Audits Guidebook, November 1990.
- GSFC Software Engineering Laboratory, SEL-84-101, Manager's Handbook For Software Development, November 1990.
- GSFC Software Engineering Laboratory, SEL-81-305, Recommended Approach to Software Development, June 1992.
- Guidelines for Standard Payload Assurance Requirements (SPAR) for GSFC Orbital Projects, Change 3, May 1992.

3.0 PURPOSE AND DESCRIPTION OF PROJECT SOFTWARE

4.0 RESOURCES, BUDGETS, SCHEDULES, AND ORGANIZATIONS

This section describes the business aspects of the Project's software acquisition. It shows how the Project will manage cost and schedule, and specifies requirements to be levied on software providers to develop and submit business related information.

4.1 Business Practices Definition and Revision Process

The Project has defined processes to estimate costs and schedules, and requirements for providers to estimate costs and schedules. The Project estimates will be used to validate those of providers; negotiated provider costs and schedules will be part of the arrangements between the Project and its providers. The project requires progress reporting from each provider that includes expenditures and progress. These reports will be used by the Project to manage the business related aspects of the software acquisition process and to assess progress.

4.1.1 Definition of Activities

The activities that the Project will use to manage the business related aspects of each providers are:

- Estimation and re-estimation of cost and schedule.
- Tracking of costs.
- Assessment of progress and schedule.

4.1.2 Method and Approach

The Project will develop estimates of cost and schedule for each provider, based on the Work Breakdown Structure (WBS) described in section 4.2, to compare with and validate the initial estimates received from each software provider. In addition, the Project will prepare a master schedule, to be used by each provider, for delivery of all products. The Project will, after negotiation with the provider, agree on a WBS breakdown of cost and on schedules. The Project will assess progress against the agreed to WBS resource use profiles and schedules by evaluating provider progress reports as defined in section 4.1.3. At each major life cycle review, new estimates of cost and schedule to complete will be submitted for Project review and approval. The accepted estimates will be used for assessment during the next phase of development. Deviations from estimates will result in Project action to analyze the cause of the deviation and the corrective action required.

The Project will use the reports defined in section 4.1.3 for monitoring the progress of providers of software. Problems identified will be addressed with the providers. Solutions to problems may result in changes processed through the

configuration management system, or in technical direction to the provider. Problems whose solution requires contractual changes will be dealt with through the appropriate processes via the Contracting Officer.

4.1.2.1 Initial and Revised Estimates of Resources and Schedule

Each software provider shall use a a consistent and repeatable methodology to develop, allocate, analyze, and revise software development staff hours and skill mix and schedule estimates. Providers shall document in their SMP the definition of the methodology and parameters used to produce the estimates contained in their development plans. If their methodology's estimating models and planning criteria are industry standards, or are commonly used and well documented in the technical literature, providers need only identify them. If the planning methodology used is unique or proprietary, it must be described in sufficient detail to enable the Project to emulate it and to authenticate that emulation using the provider's parameters. The SMP shall also include all non-proprietary data parameters input to the methodology/model to produce the resulting estimates and schedules.

The provider's SMP shall include initial estimates of code size for each deliverable CSCI and each non-deliverable software component to be developed under the contract. For each deliverable and non-deliverable, the required estimates are (a) anticipated total lines of code and (b) anticipated lines of new code.

The provider shall also identify deliverable CSCIs and non-deliverable software components, if any, that include common units of software. In each instance, the estimated total number of common lines of code and the estimated number of common new lines of code are to be identified.

Revisions to the code size estimates shall be provided by the provider at the Software Specifications Review (SSR), the Preliminary Design Review (PDR), and the Critical Design Review (CDR).

4.1.2.2 Progress Assessment

The Project will assess progress of each provider against both master schedules provided by the Project and against detailed schedules supplied by the provider. Progress will be assessed monthly, and problems and risks will be discussed with the software provider and corrective action agreed upon where required.

The provider shall establish and use a procedure for quantitatively measuring and reporting software development progress. The program shall consist of three elements; (a) a scheme which assigns numerical progress values (NPVs) to

development achievements, (b) a set of procedures for awarding value to products, and (c) mechanisms for documenting and tracking the quantitative status of each Computer Software Configuration Item (CSCI) and Computer Software Component (CSC). The quantitative progress assessment program and its component elements shall be completely defined within the provider's SMP.

The provider's quantification scheme shall establish the maximum numerical value (NPV) that can be achieved by the successful development of each CSCI. This maximum value shall be prorated across the life cycle products of the CSCI. The prorating scheme shall be further extended downward by sub allocation of values to second and third (if they exist) lower level CSCI elements (CSCs, modules, routines, etc.). If more than one iteration of the software life cycle (i.e. build process) is required for a CSCI, the provider's quantification scheme shall provide a simple and unambiguous means for assigning values to each build sequence. The values thus allocated and sub allocated shall be further subdivided across states of implementation (e.g. initiated, in-progress, completed, tested, accepted) through which each life cycle product and lower level CSCI elements normally proceed.

The provider's quantification scheme shall heavily weight early life cycle products (i.e. concept, requirements and design) and products achieving baseline status. The scheme shall also include penalty (negative) values for unresolved problems reported against a product that has been established as a baseline.

The provider's procedures for awarding NPVs to products shall be based on documents that officially record the results of formal inspections, reviews, audits, tests and reports. A progress summary file shall be maintained in the Software Development folder (see section 5,4,7,1,10) for each CSCI and CSC. Each file shall identify the subject component by name, identify its current implementation state and present earned NPV. The file shall also include a list of the documents that are the basis of each value awarded to the subject component. These files and the documents referenced by them shall be available to the Project.

4.1.2.3 Management Reviews

Each formal review required by Section 8.2 of this plan shall include a Software Management Review. The Software Management Review shall address the current status of the provider's software accomplishments. The review shall present accomplishment as measured by the provider's earned value system in light of planned and actual expended staff-hours, available resources and schedules. If the actual NPV, resource profile, or schedule are more than fifteen percent (15%) out of line with those established by the SMP for the current point in the development process, the provider's presentation shall show how

schedule and/or resource shortfalls are to be recovered. The software provider shall provide new estimates of cost and schedule to complete for Project review and approval. The accepted estimates will be used for assessment during the next phase of development.

4.1.3 Reporting, Monitoring, and Revision

Each provider shall report cost, progress, and schedule as specified in the arrangements with the provider. For those providers who are under a contract that includes a required Project Measurement System (PMS) and 533 reporting, the PMS and 533's will be used by the Project to extract the needed tracking information. Those providers with whom the arrangement is non-contractual or where the contract does not include a requirements for both PMS and 533 reporting, but who are being funded by the Project will cost information monthly, as specified in the provider agreement.

4.1.3.1 Monthly Reports

The provider shall routinely prepare and forward monthly software management and status reports to the Project. Each set of monthly reports shall be forwarded by a transmittal letter which lists the attached reports, identifies any reports due but not being forwarded and a statement summarizing the status of provider software activities. These reports will be reviewed and analyzed by the Project to independently assess provider progress and status of Level 1 CSCIs.

Software management and status reports shall be forwarded to the Project not later than the 10th working day of the month following the period covered by the report. The specific reports to be delivered monthly to the Project are:

CSCI and CSC Status Report - This report summarizes the current status of each CSCI and its level 2 CSCs. For each CSCI's level 2 CSC, the report shall include a status assessment statement, the CST's current NPV, and the number of problems and/or discrepancies remaining open. Copies of all software problem reports opened and/or closed during the reporting period shall accompany this report.

Performance Measurement Report - This report is a detail companion to the provider's CSCI And CSC Status Report. It includes copies of schedules that reflect actual versus scheduled accomplishments. For each CSCI and its Level 2 CSCs, this report shall also include comparisons between current NPVs and resource expenditures with those projected for the reporting period by the SMP. The cumulative number of problems opened, closed and remaining open for each CSCI and its CSCs shall also be included in the report. The report will explain major departures from planned accomplishments and expenditures, significant increases in the number of problem reports opened,

and open problem reports carried forward from the previous reporting period.

Lessons-Learned Report - This report describes unanticipated problems encountered by the provider, solutions to problems, expediting techniques/methods used, and actions taken to prevent recurrence of the problems. If no reportable lessons have been learned during the reporting period, the monthly report transmittal letter will state that no Lessons-Learned Reports are being forwarded.

4.1.3.2 Other Reports

Software Audit Report - This report provides the status of the provider's software audit activities. It shall identify the areas and items audited and the findings and action items resulting from the audit. Provider Software Audit Reports are to be forwarded to the Project not later than ten working days following the completion of the audit.

Software Review Report - This report provides a description of the formal software reviews conducted by the provider. It includes the topics covered by the review and action items resulting from the review. Provider Review Reports are to be forwarded to the Project not later than ten working days following the completion of the subject review.

4.2 Work Breakdown Structure (WBS)

The Project software WBS is shown in Figure FIG1. Each provider shall use the Project WBS as a framework for staffing and managing the software development effort. The top level elements in the WBS are as follows:

- Planning and Management - includes the development and administration of all software planning documents, management and control board meetings, and management audits.
- Acquisition/Purchase - includes the development of all procurement and purchase documents, source selection activities, and acceptance testing of commercial off-the-shelf (COTS) software.
- Analysis and Design - includes all conceptual engineering; encompassing development of "throw away" prototypes, supporting analysis, and the development of design documents including requirements definition.
- Development - activities associated with the production and control of computer code including modification or enhancement of inherited, purchased or government furnished software and operating the software development library (SDL).

- Performance Assurance - with the exception of Verification and Validation (see 4.1.1H below), includes all software assurance functions as defined in the NASA Software Assurance Standard, NASA-STD-2201-93, and as required by section 8 of this plan.

- Operation and Maintenance - activities required to utilize and correct operational deficiencies of software that has been established as an operational baseline.

- Logistics and Administrative Support - activities required to acquire and distribute software development supplies and materials; activities required to maintain and operate equipment and facilities used by the development staff. Includes transportation of equipment and staff travel.

- Verification and Validation - includes the activities identified and defined by section 8.2 of this plan.

4.2.1 Activity Definition

The provider shall use the project WBS in the process of preparation of cost estimates. Where needed, the provider shall add and define sub-activities. The sub-activities and the cost accounts definitions required by section 4.2.2 of the DID will be used as part of the cost estimation process and will be supplied to the Project as required in procurement documents. The information need not be in the provider's SMP. Therefore, this section and section 4.2.2 may be marked N/A in the provider's SMP.

4.2.2 Cost Account Definition

N/A - see above.

4.3 Resource Estimation and Allocation to WBS

The provider's SMP shall contain a matrix which shows development staff hours allocated to each WBS element for each life cycle phase.

4.3.1 Schedules

The master milestone schedule for the Project is included by reference to TBS2. The provider shall develop a master schedule, for all phases of development that concurs with the Project master schedule. The provider's software development schedule shall include all major life cycle milestone events. Section 4.1.2.1 discusses the initial and revised estimates.

The Project will maintain two levels of schedule. The Master Software Schedule contains major Project milestone events and software provider reviews and deliveries that have been

established by agreement with the provider. The Project Software Management Schedule shows each provider's schedule for the next lower level events required to meet the Master Software Schedule. The Project will monitor progress based on the Software Management Schedule and copies of the provider's updated schedules that accompany each monthly Performance Measurement Report (Section 4.1.3.1, above).

Providers shall maintain a hierarchical set of software schedules that are consistent with the Project's WBS and the Project Master Schedule. The provider's schedules shall show the activities and events required to developed each top level CSCI broken down to two week intervals. Changes that impact the Software Management Schedule but not the Master Software Schedule shall be included in the provider's monthly management reports.

Proposed changes that impact the Master Software Schedule will be controlled by the Project as Class I change requests. Such requests shall be submitted for Project review and approval at least six weeks prior to the earliest event that may be impacted by the proposed change. Provider schedules revised to include a Class I change shall be delivered to the Project one week following the change's approval.

4.3.2 Funds and Budgets

N/A

4.3.3 Organization

The Project's organizational structure is shown in Figure FIG2. It shows the Project members with major software responsibilities. Their responsibilities are as follows:

4.3.3.1 Software Manager (SM):

The SM is responsible for the successful management of the Project's acquisition of software that meets requirements and is delivered on schedule and within budget. These responsibilities include the development and maintenance of this document, i.e. the Project's Software Management Plan. The responsibilities of the SM also include, but are not restricted to, the following:

- The SM will be the Project's approval authority for all matters pertaining to the acquisition of software.
- The SM will review and approve provider's software management plan.
- The SM will ensure that at the conclusion of each life cycle phase, software size, effort, and schedule re-estimations are made and analyzed.

- The SM will serve as chairperson at all life cycle phase transition reviews. The SM will ensure that all review items are resolved.

- As a result of the review, the SM will determine that each life cycle phase has been successfully completed. If so, the SM will direct the provider to begin the work of the next phase.

- The SM will monitor provider staffing and staff changes to ensure continuity and sufficiency of expertise to meet schedule requirements.

- The SM will review progress reports from providers (as specified in section 4.1.3) . The reports will present current status, accomplishments for the reporting period, planned achievements for the next period, and issues, problems and concerns. Using the information in the reports, the SM will identify software management problems to be resolved with the providers.

- The SM will monitor the products and processes of any provider's software subcontractors to ensure end-to-end quality. Management of subcontractors will be a prime contractor (provider) responsibility.

- The SM will assure that provider software is delivered in accordance with the Project Master Schedule.

- The SM will chair the Interface Working Group (see section 5.4.7.2) which is responsible for the development and maintenance of ICD's for external interfaces.

- The SM will provide technical direction to software providers and support contractors especially on issues which potentially have long-term effects on system schedule and cost.

- The SM will chair the Software Risk Management Review Board, (See Section 9) and will manage risk reduction processes, based on the assessments of the Board. The SM will approve or disapprove waiver requests submitted by providers and assessed by the Board.

4.3.3.2 Software Assurance Manager (SAM) Responsibilities

The Project SAM is responsible to the SM for ensuring that provider software management, development, and assurance programs are being conducted according to the provider's approved software management and development plan and the applicable standards and procedures.

The SAM's responsibilities include but are not limited to:

- Establish the Project's software assurance requirements and procedures.

- Plan and conduct the Project's software assurance program.

- Review the Software Assurance sections of provider Software Management Plans and recommend changes and/or approval.

- Assure that all software capabilities are being developed or procured according to the providers software management plan by:

  - Evaluation of providers' software assurance activities by review of their Software Assurance Reports.

  - Conduct of scheduled and unscheduled software audits, participation in software management meetings and software technical reviews, assessment of software reports and data.

  - Witnessing inspections and tests to assure that they are performed according to approved plans, standards and procedures.

  - Examining the results of reviews, inspections, and tests to verify that the products meet their acceptance criteria.

- Assure that all software products are adequately reviewed and/or tested for compliance with established standards and requirements.

- Assure that reviewed documents and tested software are the current, correct versions.

- Assure that the Project's Nonconformance Reporting and Corrective Action (NRCA) data base is established and kept current. and that all non-conformance are properly documented and entered.

- Assure that all changes to the software are made in accordance with approved software configuration management procedures.

4.3.3.3 Software Configuration Management Officer (CMO) Responsibilities

The Project CMO, as head of the Project's Software Configuration Management (SCM) organization, is responsible to the Software Manager for establishing and maintaining the proper level of Project control over its products. Specific CMO responsibilities are to:

- Establish the Project's Configuration Management (CM) system and provider requirements.

- Establish and maintain the Project's Change Request (CR) tracking data base.

- Develop the Project's Software Configuration Management Plan section of the SMP.

- Manage the SCM library and thereby control the use and revision of official copies of baseline components.

- Act as secretary to the Project's Configuration Control Board (CCB) by preparing and distributing its agendas and minutes, recording status of CRs effected by CCB deliberations and preparing Project change authorizations for CCB approved CRs.

- Produce and distribute periodic CR data base and individual product CR status reports.

- Support Project functional and physical configuration audits (FCA & PCA) of providers.

- Review provider's Configuration Management Plan for conformance to Project requirements.

4.3.3.4 Provider Organizational Requirements

Provider's software activities shall be organized and structured such that their management interfaces with the Project, hardware and service providers, and one another are appropriate in kind and scope of authority.

In order to interface with the Project organization shown above, and to carry out the responsibilities of developing software, each provider organization shall designate:

- A qualified software specialist to act as its highest level Software Manager (SM) for all Project related software development functions. The provider's SM shall be responsible for planning and directing all aspects of software developments, acquisitions, subcontracting, products and services. The SM shall be identified in the provider's Software Management Plan by name, title and organizational placement. Should the provider choose to further delegate responsibilities, lower level managers shall report to the provider's SM and they shall also be identified in the SMP.

- A qualified software specialist to act as Software Assurance Manager (SAM) for all Project related software activities. The SAM shall have a reporting channel to management of the provider's organization that is independent of the

provider's Project management and software development function.

- A Configuration Management Officer to manage and direct the provider's configuration management process.

- An Independent Software Test Group (ISTG) for all software testing except unit level development testing. Although the ISTG shall not include persons involved in the development of the software; members of the development activity may participate in testing in a supporting role.

4.3.4 Equipment

All equipment required to support the development of Project software shall be listed in the provider's SMP. Section 5.4.2.1.2 contains requirements for a Software Support Environment (SEE) that includes CASE tools. Equipment for the SEE shall be listed.

4.3.5 Materials, Facilities, and other Resources

All materials, facilities, and other resources, including the software portions of the SEE, required to support the development of the Project software shall be listed and described in the provider's SMP.

4.3.6 Management Reserves

N/A

4.4 Work Authorization

The agreement with each provider shall serve as an authorization to proceed with work as documented in the agreement, subject to any special contract requirements for other authorization processes. For work that is in addition to an original contract, changes to the software, its interfaces, cost, and schedule shall be processed and assessed through the provider's CM system and then the Project's CM system as required in Section 10 and then appropriate documents shall be forwarded to the Contracting Office for official direction to the contractor. Only the Contracting Officer may direct the commitment of Government funds and authorize additional work.

For GSFC internal providers who are working under an agreement other than a contract, the SM shall authorize additional work or additional funds.

5.0 ACQUISITION ACTIVITIES PLAN

This section of the SMP describes how the Project will effectively manage the activities of software providers. It identifies software management requirements and constraints that are binding upon software providers. Standards that providers are to use in the development and assurance of software products are also specified.

5.1 Procurement Activities Planning

Procurement planning information that supports the Project's acquisition of software by means of competitive contracts is sensitive data and must be controlled. It is inappropriate to include such information in a SMP that will be widely distributed and reviewed in public prior to the procurement. The topics of information identified by this section are included in controlled documents that are required by NASA's procurement policies and procedures. These sensitive documents are available only to authorized individuals.

5.1.1 Procurement Package Preparation

N/A

5.1.2 Proposal Evaluation

N/A

5.1.3 Contract Negotiation

N/A

5.1.4 Procurement Risks

N/A

5.2 Organizational Requirements and Life Cycle Adaptations

5.2.1 Business Practices, Resources, and Organizational Requirements

See section 4.1. for business practices and resources and section 4.3.3 for Project organizational information and provider requirements.

5.2.2 Life Cycle Adaptations and Approved Waivers

Providers shall use the NASA Software Acquisition Life Cycle, Release 4.3. Provider proposed adaptations to the life cycle, such as development by builds, incremental development and/or phased delivery shall be described in the provider's SMP. In proposing any adaptation, the provider shall describe the

reviews and their relationships to the life cycle phases, and the baselines to be struck at the completion of the reviews.

In addition to the reviews in the life cycle (discussed in section 8.2) the
the provider shall conduct a Software Management Review (SMR) immediately following each life cycle milestone review. SMRs shall address the technical and programmatic risks associated with the next phase of the life cycle and the provider's risk management plans.

A detailed discussion of the life cycle steps is given in section 5.4 and its subsections.

5.3 Management Approach

The Project's approach to managing the development of Project software is, in priority order, based upon ensuring (a) that critical functional, performance and quality requirements are satisfied, (b) that effective use of development resources is maximized, and (c) that delivery schedules are met. The responsibility of ensuring that the management objectives is met is assigned to the software manager.

5.3.1 Software Management Responsibilities

The following table lists the CSCIs shown in Section 3.0 and the provider for each:

CSCI	Provider

5.3.2 Categorization and Classification Policy

It is the Project's policy that development of software units shall be carried out using management, engineering, and assurance practices that are appropriate to the level of cost and risk inherent to development and use of the unit and its potential impact upon the software system and the Project if the unit fails to fulfill its requirements.

Based on the Project's policy and the software risk categories defined below, providers shall describe within their SMP a process for determination of the risk categories of software to be developed and the management, engineering and assurance practices to be associated with each category.

5.3.2.1 Software Categories

The following software risk categories have been established by GSFC Management Instruction (GMI):

Category A - Critical Software

Software categorized as "Critical" is required to be highly reliable and of high quality. It will have to meet rigorous operational scenarios, and the consequences of failure are high. Full application of state-of-the practice software management, engineering, and assurance techniques are required to assure the software will fulfill its assigned role.

Category B - Important Software

Software categorized as "Important" is required to be above normal in reliability and quality. It is a key part of a system the failure of which could cause the loss of a difficult to replace asset or allow unauthorized access to data covered by the privacy act. Use of formal, high level software management, engineering, and assurance practices are required to meet the reliability and quality needs for the software.

Category C - Normal Software

Software categorized as "Normal" is expected to operate reliably. Occasional failures can be tolerated, and failure of the software system cannot cause loss of a NASA asset. Adequate methods are to be in place to detect failures of the software and the effects of such failures and to compensate for them. Formality and organization are expected in the management, engineering and assurance processes used to develop the software, but extensive efforts are not to be applied to increase the reliability of the system.

Category D - Limited Use Software

Software categorized as "Limited Use" is generally personal in nature. It is acceptable if it fails rather frequently, and other qualities, such as ease of change, may be more important than reliability. Conservation of resources during development is determined to be more important than assuring quality and reliability, and there is a correspondingly low level of formality in the use of management, engineering, and assurance practices.

CSCIs to be developed are defined in section 3. The CSCIs are assigned to the above criticality categories as follows:

CSCI	Provider	Category

5.3.2.2 Application

The provider shall define management, engineering, and assurance activities for each category to which the Project has assigned their CSCI that are appropriate to the criticality of the category. The applicable activities shall be used during the

Requirements and Architectural Design Phases of the life cycle phases. Following the PDR for the CSCI, providers shall categorize each lower level component of the design architecture. In this process, the lower level components shall be assigned to one of the categories in section 5.3.2.1, but in no case to a category higher than that assigned to the CSCI of which the component is a part unless the CSCI is reassigned to the higher category. Providers shall update their Software Management Plans to include these categorizations and the rational for each assignment. The revised provider SMP shall be transmitted to the Project for approval.

After assignment of the lower level components to categories, the provider shall apply the appropriate management, engineering, and assurance activities to each component. Aggregates of components shall be treated at the criticality level of the highest component in the aggregate.

5.3.3 Management Mechanisms

The following paragraphs identify the mechanisms that the provider shall use to control software life cycle development activities.

5.3.3.1 Requirements Development and Control

First level system software requirements are defined by TBS1. The provider shall derive lower level requirements from the requirements in TBS1 using the processes identified by Section 5.4.3.

The Project Configuration Control Board (CCB) and provider's CCBs are responsible for controlling software requirements that have been established as baselines at their respective levels. The control processes shall be in conformance with the CM requirements in Section 10 of this document.

Detailed requirements for each CSCI shall be documented according to NASA-DID-P200. Software providers shall furnish the Project with draft copies of the CSCI's preliminary, final and revised Requirements Document prior to the CSCI's SCR, SRR and PDR respectively.

5.3.3.2 Schedule Development and Control

See section 4.3.1 for schedule development and control processes and provider requirements.

5.3.3.3 Resource Development and Control

See section 4.1 for resource development and control processes and provider requirements.

5.3.3.4 Internal Review Concepts

N/A

5.3.3.5 External Review Concepts

The Project will conduct formal reviews of the provider software as described in Section 8.2 of this document. Provider requirements for formal reviews are also defined in Section 8.2.

5.3.3.6 Board Support

The Project has established a software CCB. Its functions and roles are as defined in section 10. In addition, the Project has established an Interface Working Group. Its role and functions are as described in Section 5.4.7.2. The Software Risk Management Board will support the software manager in analysis and control of risks. Its functions are described in Section 9 and are refered to in Section 5.3.8.

5.3.3.7 Management and Control

Control of costs and schedules and assessment of progress is explained in Section 4. Risk management processes to be followed are in Section 9. Configuration Management is in Section 10.

5.3.3.8 Metrics

The Project will use metrics as management and quality indicators. To support this use, each provider shall establish and implement a software metrics program which will enhance their capabilities to manage and direct the software development process and facilitate the growth of product quality.

Software metric data shall be collected that support the quantitative evaluation and analysis of trends for the entire life cycle development process and the products that it generates. Metrics to be collected include, but are not limited to:

- Number of requirements established/modified/deleted
- Software change requests
- Source lines of code estimates
- Design/code complexity index
- Percent memory, CPU, and I/O utilization
- Source code growth rate
- Detected code error rates
- Problem reports opened/closed/remaining open/cumulative
- Effort data (staffing profile)
- Development CPU time usage and trends
- Number of audits, inspections, reviews, walk-throughs, etc.
- Development activity status

The collection, reporting and analysis of metrics shall be automated to the fullest extent practicable and shall be performed on a monthly basis.

Metrics shall be provided to the Project both as raw data and in graphical form.

5.3.4 Documentation Requirements

The documentation requirements placed on providers are shown in Appendix 1.

5.3.5 Risk Management

See section 9 of this document for the Project's Risk Management approach and provider requirements.

5.3.6 Configuration Management

See section 10 of this document for the Project's Configuration Management approach and provider requirements.

5.3.7 System Assurance and Integration

See Section 8 of this document for the Project's Assurance plan and provider requirements.

5.3.8 Deviation and Waiver Procedures

The Project will review all provider requests for deviations and/or waivers to software standards and requirements. Waiver requests shall be submitted in writing to the Project Software Manager, explaining the circumstances for the request and the justification for it. The waiver request shall be submitted and approved before the provider takes any action based on the waiver. The Project Software Risk Management Review Board will review the waiver and advise the Software Manager of its assessments of the risks contained in granting the waiver. Waivers may only be granted by approval of the Software Manager.

5.3.9 Maintenance Of Management Plan

The Project SM will review and revise this document according to the schedule given in Section 1.4.

The provider's SMP shall be maintained to be current throughout the software life cycle by incorporating those changes resulting from milestone reviews and risk abatement decisions. Revisions to the provider SMP are to be presented during the Software Management Review session of the next formal review that follows the revision.

5.4 Technical Approach

5.4.1 System Requirements and Constraints

System requirements and constraints are described in TBS1. This document is the baseline from which the software provider will begin the development process.

5.4.2 Integrated System Description

The software system is described in section 3.0 of this plan.

5.4.3 Software Requirements Definition Process

The software requirements for the Project were developed during the Project's software concept and initialization phase. Both the requirements and concept documents are available to software providers for use in developing specific and detailed software requirements for their software. Project software requirements are contained in TBS1. This document is controlled at the Project level, and any changes to these requirements can only be done as a result of an approved Change Request, processed as explained in section 10 of this plan.

During the software requirements phase, the software concept and allocated system requirements are to be analyzed and documented as software requirements. Software requirements are to be documented in software requirements documents in the format given in NASA -STD-2100-91, DID P200. Each software requirement is to be identified with a control number. The requirement document shall show the source of each requirement, cross referencing to the Project provided documents.

The provider shall describe, in his Software Management Plan, the processes to be used to analyze the Project level requirements to produce the detailed requirements from which design and testing can be done. The process may include modeling, simulation, and prototyping as appropriate. Detailed requirements shall be identified as functional, performance, and interface requirements.

As a part of the requirements analysis, test planning is to begin, with a general method for verifying each requirement identified and included in a preliminary test plan. A general method of testing shall be identified for each numbered requirement.

As the requirements analysis is done, the provider shall identify technical risks and shall establish risk management control mechanisms.

The requirements are reviewed in the phase ending Software Requirements Review (SRR) (see section 8.2). A software requirements baseline shall be established by the provider after the satisfactory resolution of issues raised at the review.

The contents of the software requirements baseline become a permanent part of all succeeding baselines and are the basis against which the remaining development effort is authenticated.

5.4.4 Software Design and Implementation Process

The software design and implementation process involves completing the software preliminary design, the software detailed design, the coding and unit testing, and the integration of all software modules. The process is to be done in four phases, each of which is described below.

5.4.4.1 Software Architectural Design Phase

The objective of the software architectural design phase is to develop an overall design for the software, allocating all of the requirements to software components. The software requirements are controlled and managed, and the contents of the requirements baseline are changed only by a formal process. The phase ends with the preliminary design review, during which the acquirer and developer agree on the architecture of the system that is to be produced. Rework and action items resulting from the review are tracked and completed.

The software provider shall group the requirements into logical sets that contain those that are to be satisfied by a design unit or Computer Software Configuration Item (CSCI). As a general rule, a CSCI is to be established for a separable piece of the software system that can be designed, implemented, and operated independently. Other criteria that may go into the decision to manage a piece of software as a CSCI are:

- The software is critical to the overall performance, or there is a high level of risk involved, or system safety related tasks are contained in the item.
- The software is highly complex, incorporates new technologies, or has stringent performance requirements.
- The software encapsulates interfaces with other software items that currently exist or are provided by other organizations.
- This part of the software is expected to have more than usual change or modification after it becomes operational.
- The software contains all of a specific domain of functionality such as application, operating system, etc.
- The software is installed on a different computer platform from other parts of the system.
- A part of the software is planned to be reused.

The preliminary design shall show the design of the software to at least the functionality of the software at the next level down the control hierarchy from the CSCI. It shall be documented according to DID P300.

Each provider shall describe in the Software Management Plan the methods and processes to be used to do the detailed design. During the process, the provider shall identify risks based on the design and shall update his risk management control processes to deal with all identified risks.

The Project will review the software preliminary design at a formal Software Preliminary (Architectural) Design Review (PDR) (see section 8.2).

A Software Allocated baseline, shall be established after the completion of the PDR. The Allocated baseline shall contain the architectural design of the system and documents showing how the requirements are allocated to the design. It shall also contain all the updated documents from the Requirements baseline, along with the architectural design specification.

5.4.4.2 Software Detailed Design

During the software detailed design phase, the architectural design is expanded to the unit level. The resulting detailed design defines the design of each CSCI in a way that will provide all the capabilities and meet the deign constraints specified in the software allocated baseline. Software specifications include designs at a level and in a form that such that unit design, coding, and testing can be performed. This specification identifies the modules that make up the CSCI, the architecture of each module to the unit level, the module and unit interfaces, the data files to be used during the execution of the CSCI, and the user interface to be implemented in the CSCI. The detailed design shall be documented in the software detailed design specification, according to DID P400.

The provider shall describe in the Software Management Plan the activities that will be carried out and the methods to be used during the detailed design phase. During the process, the provider shall identify risks based on the design and shall update his risk management control processes to deal with all identified risks. During the phase, interface control documents shall be completed and test plans shall be revised. Constraints and object system resource limits shall be re-estimated and analyzed.

At the end of the phase, the Project will conduct a Software Critical Design Review (CDR). After completion of the review and the resolution of issues raised at it, the provider shall establish a software design baseline. This baseline contains the detailed (code to) design for the software.

5.4.4.3 Software Implementation

During the software implementation phase, the software is coded and unit tested. All documentation is produced in quasi-final form, including internal code documentation. At the end of the phase, all required products shall be ready for delivery, subject to modification during integration and testing.

The provider shall describe in the Software Management Plan the processes to be used during the software implementation process.

At the end of this phase, the Code Baseline is struck. This is the first time that the code itself is becomes part of a configuration management baseline. This baseline shall be a provider baseline, without delivery and acceptance review of products by the Project.

5.4.4.4 Software Integration and Test Phase

The objectives of the software integration and test phase are to integrate the software units into a completed system, discover and correct any nonconformances, and prepare for the formal acceptance of the system. The phase ending review is the test readiness review, during which the developer provides to the acquirer evidence that the software system is ready for acceptance testing. During this phase, the test plan is executed, the software product documentation is updated and completed, and the products are finalized for delivery.

The provider shall describe in the Software Management Plan the methods and procedures to be followed in the integration and test process. The plan shall show how the provider testing organization will use the code baseline, which shall include baselined test plans, to test and integrate the CSCIs and then to integrate them into a deliverable software system. The plan shall also describe how, after the controlled software components have been integrated and tested, the integrated software will be placed under configuration management control in a program library.

After the system testing has been completed and put under formal control, an FCA shall be performed to authenticate that the actual performance of each CSCI complies with the requirements stated in the baselined software requirements document. This is accomplished by evaluation of the test methods, procedures, reports, and other engineering and design documentation.

After the FCA has been successfully completed, a PCA shall be conducted to examine the as-built CSCI against required deliverables, as defined in the contract deliverables requirements list (CDRL). The PCA is performed to ensure that all deliverable items are present and complete, and the system is ready for acceptance testing.

After the provider certifies that the FCA and PCA are complete, the Project will conduct a Test Readiness Review (TRR). After resolution of any problems found during the TRR, the software integrated baseline is struck. This baseline contains the deliverable software and documents, updated to show as built design. Along with the software, all other deliverable items, such as populated data bases and tables, computer installation procedure, and test beds are part of this baseline.

5.4.5 Software Test and Delivery Process

During the software acceptance test and delivery phase, the formal acceptance procedures are carried out. The Project will witness a requirements-driven demonstration of the software to show that it meets the baselined requirements. In addition, the Project will tests the software, using requirements driven tests based on operational scenarios. These tests will be prepared by the operations staff. It is the intent of this testing to assure that the software will function correctly in its intended environment. At the end of the phase, a software acceptance review and audits (FCA and PCA) will be conducted by the Project.

The provider shall describe in the Software Management Plan how it will support the Project's activities as described above. The provider shall conduct the demonstration and shall support the test process. The provider shall resolve all nonconformances identified during the demonstrations, the Project testing, and the FCA and PCA.

At the end of the testing and any retesting required by nonconformances, the Project will conduct an acceptance review. The review will consider the test results, the FCA and PCA results, and the Project QA reviews of documents and code. After resolution of issues identified at the review, the software products will be accepted for use.

The accepted products become the Product Baseline, which will be controlled by the Project as cited in the configuration management section of this plan.

5.4.6 Software Maintenance and Updating Process

After acceptance the software is used to achieve the objectives for which it was acquired. Corrections and modifications must be made to the software to sustain its operational capabilities and to upgrade its capacity to support its users. Operation and use of the software will be done by the operations provider. The provider will be contractually required to support the software, resolving discrepancies found during operations.

Software changes that are corrective action will be made by the provider. The provider shall also train the operations contractor to do corrections and modifications to the software.

The provider shall detail in its Software Management Plan the processes and activities it will follow to phase over to the operations contractor the maintenance responsibility and capability. Especially important is the phase over of the development engineering environment.

During operational use, the baselined operational software and all baselined documents are under strict configuration management control of the acquirer CMO. No baselined software and applicable documents can be changed without following the change request process, including CCB approval.

Details of the Sustaining Engineering and Operations activities of the operations contractor are shown in section 7 of this plan.

5.4.7 Software System Engineering

The following sections address requirements for software engineering methodology that are to be followed by all software providers.

5.4.7.1 Implementation Policies and Standards

The following polices and standards apply to all software providers and all software developed by them. Specific standards to be followed are listed in Section 2.2, Applicable Documents.

5.4.7.1.1 Software Development Methods

The provider shall use systematic and well-documented structured software development methods to perform requirements analysis, design, coding, integration, and testing of the software. The methods to be used shall be documented in the provider's Software Management Plan.

The provider shall implement software development methods that support formal reviews and audits. The formal reviews and audits are those summarized in sections 5.4.3 through 5.4.5, and explained in detail in section 8.0. The provider shall document software development methods and coding and style standards in the Software Management Plan.

5.4.7.1.2 Software Engineering Environment

The provider shall establish a software engineering (development) environment to support the software engineering effort. The provider shall implement plans for the installation, configuration control, and maintenance of each item in the environment. The provider shall document the software engineering environment in the Software Management Plan. This description shall include the hardware and software tools, their uses, their maintenance, the configuration control of the tools,

and the period during which they will serve as a part of the environment.

The provider shall use a computer-aided software engineering (CASE) tool(s) which implements the chosen structured analysis and design methods to produce behavioral and essential models. The CASE tool(s) selected for use must be available for purchase by the Project. The plans for the use of these tools and for what products generated by the tools are to be incorporated into software documentation shall be documented in the Software Management Plan. These products and the developed source code shall be available in electronic format to the Project upon request.

5.4.7.1.3 Top-Down Software Design

Design shall be initiated by establishing a functional design hierarchy, where the top is the overall mission to be performed by the provider's CSCI.

Downward development of design shall be performed by:

- Reviewing and expanding the functions from higher levels to each lower level with the assistance of CASE tools to verify interfaces and document the results.
- Establishing criteria (e.g. size, complexity, use of common data, etc.) for defining CSUs.
- Iteratively evaluate functions, CSU selection criteria, and design concepts to establish CSUs of the hierarchy, functions of each CSU, and interfaces.
- Reconciling differences in the software design and the requirements allocated to CSUs at each level.
- Recording the criteria, rationale, and tradeoffs used to establish the selected design in the SDFs.

5.4.7.1.4 Non-Developmental Software

To facilitate cost-effective development and support of deliverable software, the provider is encouraged to incorporate non-developmental software (commercially available software, Government-furnished software (GFS), public domain software, and proprietary software into the software design as appropriate.

The provider shall perform the following activities prior to incorporating non-developmental software into the software design:

- Describe in the Software Management Plan the data rights and documentation the provider plans to provide to NASA.

- Describe in the Software Management Plan the plan for evaluating software and documentation to determine whether such software satisfies specified system requirements and performs as documented.

- Describe in the Software Management Plan plans for meeting software security requirements.

- Document the non-developmental software in the manner required of developmental software. A reverse engineering tool may be used for integration of this software into the developmental system so that the required CASE tool documentation products may be generated for the software system as a whole. This also applies to the use of existing proprietary software.

- Document plans for the test and validation of software to be incorporated into the CSCI according to the Software Management Plan .

If non-developmental code is used to satisfy software requirements, the following requirements apply:

- The provider is responsible for the selected software meeting the functional, performance, and interface requirements placed on it.

- The provider is responsible for ensuring that the software meets all applicable standards, including those for design, code, documentation, or for securing a Project waiver to the standards.

- Acceptance testing of the GFS, existing, or purchased software shall be done in accordance with methods used for developmental software.

- GFS which does not meet the requirements placed on it, or which functions in a manner inconsistent with requirements placed on the provider, shall be promptly and formally documented and reported to the Project. The Provider shall obtain Project approval prior to making changes to GFS.

- The provider shall apply the configuration management requirements of section 10.0 and the nonconformance reporting and corrective action requirements of Section 8.7 to such software.

5.4.7.1.5 Computer Software Organization

The provider shall partition each the software into CSCIs in accordance with the description in section 5.4.4.1. Each CSCI shall be partitioned into CSCs and Computer Software Units (CSU)

in accordance with the development methods(s) documented in the provider's Software Management Plan.

5.4.7.1.6 Traceability of Requirements to Design

The provider shall maintain traceability throughout the development process. All design and interface requirements, throughout the life cycle, shall be traceable to the Project provided software requirements and to the derived software requirements. The CASE environment shall contain a traceability tool which shall be used to maintain the traceability. The tool and requirements shall be electronically accessible by the Project. The use of this tool with respect to software requirements traceability shall be documented in the provider's Software Management Plan.

The provider shall also document the traceability of the requirements allocated from the system specification to each CSCI and its CSCs and CSUs. The requirements allocated to each CSU, CSC, and CSCI shall be traceable back to the Project provided requirements document.

5.4.7.1.7 High Order Language

Software implementation shall be done in a High Order Language (HOL). The following requirements apply to use of HOLs:

- The provider shall describe the High Order Language(s) (HOLs) selected to be used and define the criteria used for selection in the Software Management Plan.

- The provider shall notify the Project and receive its agreement prior to use of any programming language different from those described in the SMP.

- If Ada is to be used for any part of the implementation, its use shall be in accordance with the NASA/GSFC Software Engineering Laboratory's Ada Style Guide and MIL-STD-1815A. The providers development plan shall incorporate the use of these standards.

- Compilers used shall meet the corresponding National Institute of Science and Technology (NIST) or American National Standards Institute (ANSI) standards.

5.4.7.1.8 Design and Coding Standards

The provider shall establish and document in the SMP design and coding standards to be used in the development of software.

5.4.7.1.9 Software Development Folders

The provider shall use Software Development Folders (SDFs) for the control and documentation of software. SDFs may be

electronically maintained in the case tool or other location. Project access to SDFs shall be available. The provider shall establish and document procedures for creating and maintaining SDFs in the SMP. The provider shall also establish and document SDF configuration control procedures in the SMP. SDFs shall meet the following requirements:

- The provider shall document the development of each CSU, CSC, and CSCI in software development folders (SDFs). The provider shall establish a separate, organized SDF for each CSU or logically related group of CSUs, each CSC or logically related group of CSCs, and each CSCI.

- The provider shall establish the folders within one month after the completion of the preliminary design review and shall maintain the SDFs until the delivery of the final product and completion of the contract.

- The SDFs shall be made available for Project review upon request. SDFs may be generated, maintained, and controlled by automated means. To reduce duplication, SDFs should not contain information provided in other documents or SDFs.

- The SDFs shall include (directly or by reference) the following information:

  - Design requirements

  - Design considerations and constraints.

  - Source code.

  - Design documentation and data

  - Schedule and status information

  - Test requirements and responsibilities.

  - Test cases, procedures, and results.

  - Configuration control activities and change reports to include traceability of the requirement for change, the authorizing authority, and the changes made to software/documentation.

  - Results of walkthroughs, reviews, and inspections, with findings and recommendations and actions taken.

5.4.7.1.10 Processing Resources and Resource Capacity

The provider shall analyze the processing resource requirements, such as computational use and timing, memory utilization, I/0 channel utilization (including bus utilization) and shall

allocate the available resources among the CSCs and CSCIs. The provider shall monitor the utilization of processing resources and shall reallocate the resources as necessary to satisfy the reserve requirements.

5.4.7.1.11 Maintainability

Software maintainability is defined as the ease of detecting and correcting errors in the software. The provider's plans for producing maintainable code shall be included in the SMP. The provider shall meet the following requirements:

- The provider shall follow the proposed coding and design standards in order to assure that code and documentation errors are easy to detect and correct.
- The provider shall maintain traceability between requirements, designs, implementations, and test procedures to facilitate error detection and correction.
- The provider shall provide plans for producing the information necessary for the operation and maintenance of the software in the sustaining engineering section of the SMP.

5.4.7.2 Interface Control Process

Interfaces external to the providers software will be documented in Interface Control Documents (ICDs). These ICDs will be developed and placed under Configuration Management control at the Project level. The Project has established an Interface Working Group (IWG), chaired by the Software Manager, to develop the ICDs. Each software provider will name a member of the working group. Schedule of completion of the ICDs is shown in TBS2.

5.4.7.3 Data Generation and Management Process

Development of default and baseline values for tables and parameters used to control the operations or computations of software shall be the responsibility of the software provider, except where the parameters are, in effect, an interface to another system or component. In the later case, the selection of the parameters and development of their values is the responsibility of the IWG. Once selected and approved by the Project, the parameters and their value shall be placed under Configuration Management.

Generation of test data shall be the responsibility of the provider. Selection and availability of test data from providers of other system components will be the responsibility of the Project Software Manager. During the requirements analysis process, a preliminary test plan is to be developed

(see section 5.4.3). Each version of test plans shall show the source of data needed to execute the tests.

5.4.7.4 Performance Assessment Process

The assessment of performance against performance requirements shall be done as part of the integration and acceptance testing described in sections 5.4.4.4 and 5.4.5, and generally referred to in section 8.2

5.4.7.5 Operations Maintenance Process

Sustaining engineering and operations are described in section 7.0 of this plan.

6.0 DEVELOPMENT ACTIVITIES PLAN

This section of the plan is omitted since the Project will not directly develop any software. Each provider shall include a Development Activities Plan in the required Software Management Plan (see section 1.2.

7.0 SUSTAINING ENGINEERING AND OPERATIONS ACTIVITIES PLAN

TBS by Project

8.0 ASSURANCE PLAN

The Project will conduct a software assurance program that shall include quality assurance, verification and validation, quality engineering, safety assurance, and security and privacy assurance. The Project Software Assurance Manager (SAM) shall be responsible for planning and execution of the assurance program. The following paragraphs detail the Project's plan, and specify software assurance requirements for software providers.

Each software provider shall conduct a software assurance program that satisfies the provider requirements in this document and that satisfies the requirements in the NASA Software Assurance Standard, NASA-STD-2201-93.

Each Provider shall include in the required software management plan (see section 1.2) a plan for a software assurance program in accordance with the above stated requirements.

8.1 Quality Assurance Planning

The Project and each provider shall conduct a program of Software Quality Assurance (SQA), which is a planned and systematic approach to the evaluation of the quality of and adherence to software product standards, processes, and procedures. SQA includes the process of assuring that standards and procedures are established and are followed throughout the software acquisition life cycle. Compliance with agreed-upon standards and procedures is evaluated through process monitoring, product evaluation, and audits. Software development and control processes shall include quality assurance approval points, where an SQA evaluation of the product shall be done in relation to applicable standards.

8.1.1 Approach and Activities

The Project will conduct oversight of the provider SQA organization to assure that the provider is carrying out a software assurance program that meets requirements. The Project SAM is assigned this oversight responsibility. As part of the oversight role, the Project will perform both scheduled and unscheduled audits of providers to establish the degree of conformance to the standards and procedures and to reported status.

The Project will review and approve the standards and procedures proposed for use by providers, and will assess the quality of all of the provider's delivered products against the appropriate standards.

The provider shall conduct software quality assurance activities throughout the software life cycle in accordance with the following requirements:

- During the software concept and initiation phase, the software quality assurance activity shall be involved in the writing and reviewing of management plans to assure that the processes, procedures, and standards identified in the plan are appropriate, clear, specific, and auditable.
- During the software requirements phase, the software quality assurance activity shall assure that the software requirements are complete, testable, and properly expressed as functional, performance, and interface requirements.
- During the software architectural (preliminary) design phase, the software quality assurance activity shall:
    - assure adherence to design standards;
    - assure that all software requirements are allocated to software components;
    - assure that a test verification matrix exists and is kept up to date;
    - assure that Interface Control Documents are in agreement with the standard in form and content;
    - review Preliminary Design Review documentation and assure that all action items are resolved; and
    - assure that the approved design is placed under configuration control.
- During the software detailed design phase, the software quality assurance activity shall:
    - assure that approved design standards are followed;
    - assure that the results of design inspections are included in the design; and
    - review Critical Design Review documentation and assure that all action items are resolved.
- During the software implementation phase, the software quality assurance activity shall audit:
    - the results of coding activities;
    - status of all deliverable items;
    - configuration management activities and the software development library;

- the nonconformance reporting and corrective action system.

- During the software integration and test phase, the software quality assurance activity shall:

  - assure readiness for certification testing;

  - assure that all tests are run according to approved test plans and procedures and that any nonconformances are reported and resolved;

  - assure that test reports are complete and correct;

  - certify that testing is complete and software and documentation are ready for delivery; and

  - participate in the Test Readiness Review and assure all action items are completed.

- During the software acceptance and delivery phase, the software quality assurance activity shall assure that final functional and physical configuration audits are conducted in accordance with Project-approved standards and procedures.

- During the software sustaining engineering and operations phase, there will be mini-development cycles to enhance or correct the software. During these development cycles, the software quality assurance activity shall conduct the appropriate phase-specific activities described above.

8.1.2 Methods and Techniques

The Project will use an audit guide and checklists to perform scheduled and unscheduled audits of the provider's software process, products, and status reports. The Project will use checklists and the provider's Project-approved standards in its detailed evaluations of each of the provider's products.

The provider shall explain, in the required SMP, the methods and techniques to be used.

8.1.3 Products

The Project will develop audit reports in accordance with Audit Report, NASA-DID-R002 for each audit conducted. The results of the audit will be conveyed to the provider so that appropriate action can be taken to correct any deficiencies found.

The provider shall describe in the required SMP the products of the SQA process. The provider shall develop audit reports in accordance with Audit Report, NASA-DID-R002 for each audit

conducted. Copies of the reports shall be furnished to the Project, as required in section 4.1.3.2, for review and action.

8.2 Verification and Validation (V&V) Planning

The Project and the provider shall conduct V&V activities, to include reviews, inspections and informal technical reviews, and testing of all deliverable products.

8.2.1 Approach and Activities

The Project will conduct, and the provider shall support, formal reviews at the end of each life cycle phase. These reviews shall include the Software Requirements Review, the Software Preliminary Design Review, the Software Critical Design Review, and the Software Test Readiness Review.

The reviews shall encompass the items to be included in the configuration management baselines to be established after the successful completion of the review. See section 10.2.1 for the minimum contents of each baseline.

After each formal review, the Project Software Manager will decide upon the readiness of the provider to begin the next development life cycle phase. The Project Software Assurance Manager will make a readiness recommendation to the Software Manager based on an assessment of status and readiness of processes, procedures and standards needed in the next phase. After completion of rework for problems found during the review and correction of any readiness problems, permission to begin the next phase will be given.

GSFC Flight Assurance Reviews are in addition to the end of phase reviews specified above. They will be conducted by the Office of Flight Assurance. These reviews will be at the system level, and software will be among the items reviewed. The provider shall support these reviews as required above for Project formal reviews.

The provider shall conduct an inspection and internal technical review program as follows:

- The provider shall conduct formal inspections of requirements, according to NASA STD-2202-93, before proceeding to design and implement the requirements.
- The provider shall conduct formal inspections or walkthroughs of all other deliverable products.

The provider shall conduct three levels of testing: unit, integration, and acceptance readiness testing. Unit, and integration testing shall be informal testing conducted by the provider.

Acceptance readiness testing shall be formal testing conducted by the provider and witnessed by the Project. The purpose of acceptance readiness testing shall be to show that the software is ready for acceptance testing by the Project. All discrepancies found during formal testing shall be entered in the provider's Nonconformance Reporting and Corrective Action (NRCA) system (see Section 8.7) and tracked until closure. Access to this data in the NRCA system shall be provided to the Project.

Test planning shall be done for all levels of testing. The provider shall submit to the Project for review and approval test plans for the formal testing, such as the acceptance readiness testing. Once a test plan is approved, the provider shall prepare test procedures according to DID A200. The procedures shall be used for the tests, and shall be available for Project review and comment.

The Project will conduct Formal Acceptance testing on delivered software, following a Test Readiness Review of the results of Acceptance readiness testing. Formal Acceptance Testing will include the generation of the system from source code, using installation procedures provided. The Project will prepare test plans, based on requirements and operations manuals, and will develop test procedures according to DID A200. Discrepancies found during Acceptance testing will be entered into the Project's NRCA system and tracked until closure. The provider shall correct all discrepancies found.

The provider shall conduct verification and validation activities throughout the software life cycle in accordance with the following:

- During the software requirements phase, verification and validation activities shall include:
  - analyzing software requirements to determine if they are consistent with, and within the scope of, system requirements.
  - assuring that the requirements are testable and capable of being satisfied.
  - conducting formal inspections of requirements.
  - creating a preliminary version of formal test plans, including a test verification matrix.
  - beginning development of test beds and test data generators.
- During the software architectural (preliminary) design phase, verification and validation activities shall include:

- updating the preliminary version of the formal test plan, including a test verification matrix;
- conducting informal technical reviews or formal inspections of the preliminary software and data base designs;

- During the software detailed design phase, verification and validation activities shall include:
  - updating the formal test plan and the test verification matrix.
  - conducting informal technical reviews or formal inspections of the detailed software and data base designs.

- During the software implementation phase, the verification and validation activities shall include:
  - walkthroughs or formal inspections of code.
  - unit testing of software and data structure units.
  - locating and correcting errors and testing the changed software.
  - development of test procedures for the next two phases.

- During the software integration and test phase, verification and validation activities shall include:
  - conducting tests in accordance with test procedures.
  - documenting test performance, test completion, and conformance of test results versus expected results.
  - providing a test report that includes a summary of nonconformances found during testing.
  - locating, recording, correcting, and retesting nonconformances.

- During the software acceptance readiness testing phase, verification and validation activities include:
  - conducting formal testing, according to the formal test plan and procedures, to demonstrate that the developed system meets its functional, performance, and interface requirements;
  - locating, recording, correcting, and retesting nonconformances

8.2.2 Methods and Techniques

For each formal review, the provider shall:

- Provide the required products for review.
- Develop and organize material for oral presentation to the Project review team. Copies of visual aids and other supporting material that are pertinent to the review shall be submitted to the Project at least 3 working days before the review.
- Support splinter meetings resulting from the review.
- Produce written responses to recommendations and action items resulting from the review.

Inspections shall be conducted according to NASA-STD-2202-93. Walkthroughs shall be conducted according to provider developed and Project approved procedures.

The provider shall conduct informal testing in accordance with Project-approved provider standards and procedures.

The provider shall conduct, and the Project will witness, formal testing in accordance with the Project-approved test plan and provider developed procedures.

8.2.3 Products

The products of Formal Inspections shall be inspection reports in accordance with the DID for Inspection Reports, NASA-DID-R003.

The results of informal reviews and walkthroughs shall be documented in the appropriate Software Development Folder (See Section 5.4.7.1.10). The provider shall summarize the results of informal reviews and walkthroughs in the Monthly Progress Report.

The results of informal testing shall be recorded in the appropriate software development folders.

The results of formal tests shall documented in accordance with Test Report, NASA-DID-R009.

Discrepancy reports shall be documented in accordance with Discrepancy (NRCA) Report, NASA-DID-R004. See section 8.7 for a discussion of Nonconformance Reporting and Corrective Action.

8.3 Quality Engineering Assurance Planning

Software Quality Engineering (SQE) is the activity that evaluates, assesses, and improves the quality of software. Software quality is often defined as the degree to which software meets requirements for reliability, maintainability, transportability, etc., as contrasted with functional, performance, and interface requirements that are satisfied as a result of software engineering.

The provider shall conduct a Software Quality Engineering program that satisfies the requirements of the Software Assurance Standard, NASA-STD-2201-93, section 3.3.2. In addition, the program shall satisfy the requirements in the remainder of 8.3.

8.3.1 Approach and Activities

Software classified in categories higher than "Normal" in reliability requirements will have additional processes conducted during development to ensure that reliability is built in. These categories, the software assigned to each category, and the activities to be conducted are described in section 5.3.2.

The provider shall collect data, analyze metrics, and use them to guide quality engineering activities. Metrics and associated requirements are described in section 5.3.3.8.

8.3.2 Methods and Techniques

Metric data shall be collected, stored in provider data bases, and provided to the Project. The Project will compute metrics and trends using PC based tools.

8.3.3 Products

The Project will develop graphs and other displays that can be used in management and risk analysis.

8.4 Safety Assurance Planning

The Project will identify safety risks that can be caused by the failure of software to perform as required and any system risks that are to be controlled by software during the baseline risk assessment process described in section 9.0. Identified safety risks will be tracked by the Project as technical risks, and risk mitigation actions will be the responsibility of the SRMB.

The provider shall conduct a software safety assurance program that satisfies the requirements of the Software Assurance Standard, NASA-STD-2201-93, section 3.3.5. In addition, the software safety program shall satisfy the requirements in the remainder of section 8.4.

8.4.1 Approach and Activities

The following activities will be performed to assure safety requirements are met:

- Safety hazards, if any, will be identified in the Project's baseline risk assessment process.
- Software requirements associated with safety hazards will be identified as critical, safety related, requirements.
- The provider shall describe in the SMP a plan to assure the safety of the system, the methodology for doing safety analyses, and the methods to be used to ensure that the software system satisfies critical, safety related, requirements.
- The methodology used by the provider shall contain a method for the tracing of safety critical requirements to software components and the identification of the component as safety critical. For identified safety critical software components, software safety activities shall be initiated to include requirements, design, and code analyses and special testing.
- The provider shall document and report on at all formal reviews:
  - the steps taken to identify actual and potential hazards.
  - the approaches used to confront, address, and neutralize hazards.
  - the use of safety engineering approaches.
- In the provider's testing program, the provider shall explicitly test all critical, safety-related requirements.

8.4.2 Methods and Techniques

The provider shall identify in the SMP the methods and techniques to be used to identify safety critical requirements and safety critical software components and the analyses and V&V methods to be used to ensure that they function as required. The provider shall include the following in the SMP:

- The provider shall have some form of criticality analysis; specification analysis; and timing, sizing, and throughput analysis that are used to identify safety critical software requirements and components.
- For identified critical components, the provider shall conduct formal inspection of the detailed requirements, the

detailed design, the code, and the test plan and procedures.

- For identified critical components, the provider shall have some form of design and code analysis and special safety testing, which focus on locating program weaknesses and identifying extreme or unexpected situations that could cause the software to fail in ways that would cause a violation of safety critical requirements.

8.4.3 Products

The provider shall submit risk analyses and safety hazard reports to the Project as required, and shall have available the results of all safety related analyses, inspections, and tests, in the SDFs of the critical components.

8.5 Security and Privacy Assurance Planning

The Project will conduct a security assessment process by considering and categorizing the sensitive information that is to be managed and controlled by the Project software. The information, including both programs and data, will be categorized according to its sensitivity. The categorization will meet the requirements contained in NMI 2410.7A, "Assuring the Security and Integrity of NASA Automated Information Systems."

Based on the categorization, the provider shall develop security requirements. The security requirements shall encompass system access control, including network access and physical access; data management and data access; environmental controls (power, air conditioning, etc.) and off-line storage; human resource security; and audit trails and usage records.

The provider shall conduct a software security and privacy assurance program that satisfies the requirements of the Software Assurance Standard, NASA-STD-2201-93, section 3.3.6. In addition, the software safety program shall satisfy the requirements in the remainder of section 8.5.

8.5.1 Approach and Activities

The provider shall conduct security assurance activities that are directed to ensuring that information being (or to be) processed by the software system and the software being developed has been assigned a proper sensitivity category as defined in NMI 2410.7A, and that the appropriate protection requirements have been developed and met in the software. In addition, security assurance activities shall include ensuring the control and protection of the software being developed and/or maintained, and of software support tools and data. A minimum security assurance program shall ensure that:

- A security evaluation has been performed.
- Security requirements have been established for the software and data being developed and/or maintained.
- Security requirements have been established for the development and/or maintenance process.
- Each software review and/or audit includes evaluation of security requirements.
- The configuration management and corrective action processes provide security for the existing software and that the change evaluation processes prevent security violations.
- Physical security for software and data is adequate.

The provider shall describe in the SMP the planned approach to meeting the security and privacy requirements.

8.5.2 Methods and Techniques

The provider shall review and analyze security and privacy requirements to include the following aspects: effective and accurate operations; protection from unauthorized alteration, disclosure, use or misuse of information processed, stored, or transmitted; maintenance of continuity of automated information support; incorporation of management and operational controls; and appropriate technical, administrative, environmental, and access safeguards.

8.5.3 Products

Results of the security review shall be provided to the Project.

8.6 Certification Planning

N/A

8.6.1 Approach and Activities

N/A

8.6.2 Methods and Techniques

N/A

8.6.3 Products

N/A

8.7 Nonconformance Reporting and Corrective Action

The Project and each software provider shall establish a Nonconformance Reporting and Corrective Action (NRCA) system, which shall provide for the recording of nonconformances, the evaluation of impact and establishing of priority, the tracking and reporting of status, and the closure after testing. A nonconformance shall be defined as a deviation of any product from its requirements or standards. Nonconformance reports shall be filed against any product in any phase of the software life cycle after a product is first approved or baselined by its developer and released for wider use. The NRCA system shall interface with the CM system in order to track the product changes and versions that result from correcting nonconformances.

8.7.1 Approach and Activities

A designated form shall be used to make the nonconformance report. The form shall contain at least the following information:

- Date and time of detection of the nonconformance.
- Error identification (report number and title).
- Reporting individual and organization.
- Individual responsible for corrective action.
- Criticality of the nonconformance.
- Statement of the nonconformance.
- Proposed fix for the nonconformance.
- Identifier of the unit of code, data, or documentation in which corrective action must be taken.
- Life cycle phase in which the nonconformance was introduced.
- Life cycle phase in which the nonconformance was detected.
- Final closure resolution.
- Date and/or version of the configuration item in which the correction will be included.
- Date on which the nonconformance is closed.

Nonconformance reports shall follow NASA-DID-R004.

8.7.2 Methods and Techniques

A nonconformance tracking and reporting system shall be used that is able to provide management reports containing error and correction status, the number of errors found per product, and the criticality of open problems. This data enables the impact of nonconformances to be evaluated so that the use of resources may be prioritized. All Nonconformance reports shall be entered and tracked by the reporting system. The Project shall have access to and use of the information in provider nonconformance systems.

Nonconformance reports shall be evaluated for criticality and level of importance. In addition, each nonconformance reports shall be evaluated to identify those that contain requirements changes disguised as nonconformances. Such reports shall be rejected and shall result in the opening of a change request. Factors to be considered in the criticality and level of importance shall include:

- The impact of not correcting the nonconformance.
- The resources required for correcting the nonconformance.
- The impact on other baselined items if the nonconformance is corrected.

If the decision is made to correct a nonconformance, there shall be procedures to control the corrective action process. Such procedures shall include follow up to ensure the nonconformance has been documented and corrected in the appropriate version of software, and to assure that adequate testing, including regression testing, is done. Provides shall document these procedures in their SMP.

8.7.3 Products

Each NRCA system shall provide access to the actual nonconformance reports, and shall provide summary and status reports that show the status of nonconformances.

9.0 RISK MANAGEMENT PLAN

The objective of the Project's risk management processes is the identification and control of risks, both technical and programmatic, that could cause the acquired software to fail to satisfy its requirements, including those related to schedule and costs. To this end, the Project is establishing a Software Risk Management Board (SRMB), chaired by the software manager, to review and disposition risk items. The providers of software are to establish risk management programs that identify and control risks and that support the activities of the Project SRMB.

The Project Software Manager will cause to be conducted and documented in TBS3 a baseline software risk assessment, based on TBS1. This assessment will:

- Identify potential risks to the Project that may arise from the planned software acquisition, development, utilization and support activities.
- Analyze the sources, if any, of each potential software related risk and forecast the possible consequences if the risk is not removed or counteracted;
- Prioritize potential risks with respect to their possible technical and programmatic impacts, the probability of those impacts occurring, and the estimated cost for precluding or abating the potential risk.
- Be presented to the Software Risk Management Board, including the methods and data used to perform the risk assessment, the results of the assessment, and recommended courses of action.
- Result in provider requirements for specific actions that will reduce the identified risks.
- Be available for providers in establishing their risk management plans.

Each provider shall establish an organized software risk management program that provides a systematic assessment and control of potential safety, security, technical, performance and schedule risks associated with the development and operational use of Project software. The provider's risk management program shall be documented the SMP.

9.1 Risk Assessment and Evaluation Process

Identification, assessment, and evaluation of risks is intended to be a continuous process. Both the Project SRMB and each provider shall continuously assess software activities for risk,

reporting to the Project any risks identified, along with suggested actions to reduce or eliminate the risk.

In addition to the continuing process, there are specific life cycle points at which a risk assessment shall be made. At the beginning of the requirements analysis phase and prior to each succeeding phase of the development life cycle the provider shall:

- Identify and elaborate the objectives for the software product and the development and management processes.
- Identify alternative methods of achieving the stated objectives.
- For objectives that do not have alternative methods of being achieved or are have a high probability of not being achieved, identify potential risks which may preclude the objective from being achieved.
- For each potential risk identified, quantify the possibility of an undesirable outcome.
- Prioritize identified risks based on the combination of their probability of an undesirable outcome and the impact severity of those outcomes upon the operational use of the software.
- Provide a report of the results of the above steps to the Project SRMB for review and approval.

In addition, each review of the software shall address risks. At each formal review, the Project appointed reviewers will be requested to identify potential technical, safety, security, resource, schedule, or cost risks; the relative magnitude of those risks and in light of the requirements that the Project software is to satisfy, what actions should be recommended.

The provider shall control high severity risks by:

- Instrumenting risk prone activities, collecting risk relevant data, and computing meaningful risk metrics.
- Establishing management procedures that increase visibility into high risk areas and facilitate timely decision making to prevent, avoid and reduce risks.
- Pinpointing risk sources, determining risk sensitivity over ranges of source parameter values, and resolving risk issues through a planned and systematic program of modeling, prototyping, phased delivery and reliability demonstrations.
- Monitoring significant risk activities closely, analyzing risk metrics to identify trends, tracking high risk products and maintaining management pressure on corrective actions.

9.2 Technical Risks

Following each End-of-Phase review and Project audit, the Project software manager and the Project Software Risk Management Board will independently assess software technical risks using products, reports and data furnished by the provider and the reports of the review panel and auditors. The results of these assessments and associated action items will be provided to the software provider, who will be required to control the risks.

The SRMB will assess all waiver requests submitted by providers and advise the SM of any risks found in granting or not granting the waiver.

9.3 Safety Risks

Potential software induced safety risks shall be identified through detailed safety analyses of software requirements, designs, and code. The traceability of the requirements through design and code into the test procedures shall be assured to eliminate safety risks arising from errors of omission. The combined use of fault tree analysis and Petri-net diagrams shall also be used to identify potential sources of safety risks. Elements of the software design that are identified as contributors to potential safety hazards shall be placed on a critical items list and subjected to in-depth analysis, redesign, stringent change control and extensive operational testing.

9.4 Security Risks

The provider shall periodically review the software development environment and its control processes and procedures to ensure that the possibility of improper alteration or loss of source documentation, data or code has a low probability of occurrence. The results of this review are to be reported to the Project.

9.5 Resource Risks

The provider shall identify resources, and their periods of availability, that are critical to satisfying the technical and delivery requirements of the Project. These resources include, but are not limited to; technical expertise, specialized equipment and unique facilities. The provider's SMP shall identify critical resources and a contingency plan that provider will implement in the event that a resource becomes unavailable during the period during which it is critical.

9.6 Schedule Risks

The provider shall maintain current schedule information that is available to the Project. In addition, the schedule shall be

re-estimated as provided in section 4.3.1, and risks shall be identified during the process.

The providers software configuration Management process shall include a process for estimation of schedule impact of proposed changes. A change to the Project master schedule can only be made by the Project CCB. See Section 10.0 of this document for more information.

9.7 Cost Risks

The provider shall report current cost and cost risks as required (see Section 4.1). In addition, the cost to complete shall be re-estimated as provided in section 4.1, and risks shall be identified during the process.

The provider's software configuration Management process shall include a process for estimation of cost impact of proposed changes. A change to the Project's cost can only be made by the Project CCB. See Section 10.0 of this document for more information.

10.0 CONFIGURATION MANAGEMENT PLAN

Software configuration management is the discipline of identifying the configuration of software at discrete points in time and systematically controlling changes to the identified configuration for the purpose of maintaining software integrity and traceability throughout the software life cycle. In order to accomplish the objective given in the above definition, there are four identified SCM functions:

- Configuration Identification: identification of the components that make up the software system and definition of its functional characteristics
- Configuration Control: control of changes to those components
- Configuration Status Accounting: reporting of status of the processing of change requests and their implementation status
- Configuration Authentication: audits to authenticate that the controlled items meet their requirements and are ready for delivery.

This Project Configuration Management (CM) plan establishes the processes the Project will use to manage the configuration items and changes to them. It contains requirements for providers of software for configuration management. The Project CM plan satisfies the requirements of GMI 8040.1.

10.1 Configuration Management Process Overview

The Project configuration management system will be managed and operated by the Project Configuration Management Officer (CMO). A description of the duties of the CMO is given in Section 4.3.3.3. The Project will establish an initial baseline, consisting of this plan, the software requirements, document TBS1, the Interface Control Documents (ICDs), and the Project Master Schedule, document TBS2. Additional baselines will be established by the providers, as required in section 10.2.1. Changes to the Project baseline are designated Level I changes, as are any changes that will impact Project costs and the master schedule. All Level I changes must be dispositioned by the Project. Classification of changes is discussed in section 10.2.2.2.2.

The Project will establish an operational baseline as software is accepted from the providers and used for operations. This operational baseline will be turned over to the operations and maintenance contractor.

The Project has established a Software Change Control Board (CCB), which is chaired by the software manager. The board will disposition all Change Requests (CRs) submitted to it. Those that require changes to the software being developed by a

provider shall be assessed for impact by the provider before Project CCB consideration. Level I changes referred to the Project by providers shall include an impact assessment before submission to the Project CCB. The Project CMO shall maintain a database of all submitted change requests and will manage the process of obtaining assessments, convening meetings of the CCB, managing the CCB agenda, recording the decisions of the board, and passing approved CRs to providers for action.

The Project CMO will review all Level II changes dispositioned by providers to assure that they are correctly classified (see Section 10.2.2.2.2 for a discussion of classification). The Project CMO will conduct and/or participate in FCAs and PCAs conducted by providers to authenticate the configurations of provider baselines.

Each provider shall plan, document in the provider SMP, and implement a configuration management plan, and shall implement detailed configuration management procedures. Each provider shall name an Configuration Management Officer to be responsible for the operation of the provider's CM process. Each provider shall establish a provider level CCB to disposition at least Level II CRs (see section 10.2.2.2.2), and if needed, lower level CCBs to disposition lower level change requests. Each provider shall establish a program library to take physical control of each baselined item. The Project will review and approve the provider CM plan as part of its review and approval of the provider's SMP.

10.2 Configuration Control Activities

Section 10.2 identifies the activities to be performed by the Project and the Provider to implement the four primary configuration management functions.

10.2.1 Configuration Identification

Configuration identification is the process of defining each baseline to be established during the software life cycle, by describing the software configuration items and associated documentation that comprise each baseline, and by recording the names, versions, and other identifiers of each component of the baseline.

CSCIs to be developed are described in Section 3 of this document. Additional CSCIs may be designated by providers in accordance with the criteria given in section 5.4.4.1. Providers shall include in their SMPs a method of designating each CSCI by the development of a numbering and naming scheme that can be used to correlate the software items with their components and their associated documentation.

The provider shall establish the following baselines, with contents as noted:

10.2.1.1 Software Requirements Baseline

The software requirements baseline is struck after the completion of the requirements analysis phase and the satisfactory resolution of issues raised at the phase ending Software Requirements Review (SRR). The requirements baseline shall contain the approved software requirements specification and interface requirements documents. It shall also contain other relevant provider management documentation such as the SMP and the first version of test plans. These documents detail the provider's approach to managing, developing, testing, assuring, and controlling the software. The baseline shall include applicable standards and procedures that will be adhered to during the development of the software.

10.2.1.2 Software Allocated Baseline

The Allocated baseline shall be struck after the completion of the preliminary design phase and the resolution of any problems raised at the Software Preliminary (Architectural) Design Review (PDR). The baseline shall contain all the updated documents from the Requirements baseline, along with the architectural design specification. The baseline shall also contain a software build (or release) plan and test plans.

10.2.1.3 Software Design Baseline

The software design baseline shall be struck after the completion of detailed design and the resolution of problems raised at the phase ending Software Critical Design Review (CDR). The software design baseline shall contain the detailed (code to) design for the software. This document shall include designs at a level and in a form that such that unit design, coding, and testing can be performed. The updated contents of the allocated baseline shall be part of this baseline, specifically to include updated test and build plans.

10.2.1.4 Software Code Baseline

At the end of the implementation phase, the Code Baseline shall be struck. It shall be struck after the software components (units) have been coded and successfully passed unit test. The units shall be transferred from the control of the provider developmental organization to the control of the provider CMO and placed under configuration management in a program library. The Code baseline may be built incrementally throughout the implementation process as each unit of code is successfully inspected and unit tested.

The Code baseline is the basis for CSCI and system integration testing. It shall include the updated Design baseline, with completed integration test plans.

10.2.1.5 Software Integrated Baseline

The provider's testing organization shall use the code baseline, which shall include baselined test plans, to test and integrate the CSCIs and then to integrate them into a deliverable system. After the controlled software components have been integrated and tested, the integrated software shall be placed under configuration management control in the program library.

The Software Integrated baseline shall contain the deliverable software and documents, updated to show as built design. Along with the software, all other deliverable items, such as populated data bases and tables, computer installation procedure, and test beds shall be part of this baseline.

10.2.1.6 Software Product Baseline

The completed Software Integrated baseline shall be used by the provider to conduct acceptance readiness testing. As defined in section 8.2, this is formal testing that is witnessed by the Project. After completion of the testing, the phase ending Test Readiness Review, and the formal audits (FCA and PCA) the Product Baseline shall be established.

The Software Product Baseline shall contain all of the elements of the Integrated Baseline, with corrections made for any nonconformances detected during the testing process. The Product Baseline shall be delivered to the Project for acceptance testing.

10.2.1.7 Software Accepted (As-Built) Baseline

The Project shall conduct a formal acceptance test on the Product Baseline, using independently developed test plans and procedures. After successful completion of this testing and correction of nonconformances, the Accepted Baseline is established. The Accepted Baseline shall contain all elements of the Product baseline with corrections.

10.2.2 Configuration Change Control

Configuration change control is the systematic process for evaluating, coordinating, and deciding on the disposition of proposed changes to the configuration items, and for implementing those changes to baselined software and associated documentation. The change control process ensures that the changes which have been initiated are classified, evaluated, approved or disapproved, documented, implemented, and verified.

10.2.2.1 Controlled Storage and Release Management

The provider shall establish a program library, and shall designate a librarian to manage and operate the program library. The program library shall provide actual storage and physical control of the contents of each baseline. The program library shall contain the official copies of all baselined items that make up the various CSCIs.

The provider program library and librarian shall be under the direction of the CMO. It shall accept documents, code, data files, and other components of baselines and shall maintain them in secure storage. It shall issue working copies to developers for authorized changes, and shall keep records and historical copies of all versions of the components of baselines. It shall make copies of baselined software for testing and distribution, and shall prepare version description documents.

The program library shall have access to computer resources for the maintenance of baselines, records, and files.

The Project will establish a Project program library, with a designated librarian that supports the Project CMO, to control the Project baseline and to accept and control products delivered from the providers.

10.2.2.2 Change Control Flow

An orderly change process is necessary to ensure that only approved changes will be implemented into any baselined document or software. Figure FIG3 shows a simple overview of the change process. The steps within the overall process shall be:

10.2.2.2.1 Initiation

The Project has established a Change Request form (see section 10.2.2.3)for use in documenting proposed changes and their disposition. Each provider shall set up a CR form containing at least the information in the Project form. Electronic forms may be used.

CR forms shall be submitted to the Project CMO for Project level changes, and to the provider CMO for provider level changes. The Project CMO receives the CR and reviews it for clarity and completeness. If the CMO determines that the CR is not complete, it is returned to the originator. Once complete, the CMO assigns the CR a unique identifier for tracking purposes and records information about the CR into the change request tracking data base.

In the SMP, each provider shall describe the tracking data base to be established and the procedures to be followed to assure its completeness and integrity.

10.2.2.2.2 Classification

Changes to software and associated documentation shall be classified according to the impact of the change and the approval authority needed. Level I shall be assigned to changes that would affect the Project level requirements, external interfaces, system cost, and/or delivery schedule. These changes can only be approved by the Project.

Level II and lower changes shall be dispositioned by the Provider, with copies of Level II changes sent to the Project for review. Level II changes shall be those that affect the interfaces between CSCIs and the allocation of functions to CSCIs, or effect component level cost and schedule. These changes generally shall be approved only by the provider Project level CCB. Lower levels of changes, for example, those that affect CSCI internal design and division of functionality, may be established by the provider and dispositioned at lower levels.

10.2.2.2.3 Change Evaluation

The Project will do analysis of the impact of each CR received, or will send them to the provider for analysis. Each provider shall analyze all CRs sent to it by the Project.

The provider Configuration Control process shall provide for analysis of changes in terms of impact to system functionality, utility, cost, and schedule. Each change shall also be analyzed for impact on software safety, reliability, maintainability, transportability, and efficiency. The analysis shall be documented in a report which shall describe the changes that would have to be made to implement the CR, the CSCIs and documents that would have to be changed, and the resources needed to make the change. The report shall be identified with the same unique identifier as the CR, and shall become part of a change package, along with the CR.

10.2.2.2.4 Change Dispositioning

Level I changes will be dispositioned by the Project CCB. Changes at Level II and below shall be dispositioned by the provider CCB. Each CCB shall evaluate the desirability of a change verses the cost of the change, as described in the analysis report. The CCB shall approve, disapprove, or defer a change request.

Dispositioned items shall be sent to the CMO for action and for recording of the disposition. The originator of the CR shall be notified of the disposition made by the CCB. Rejected CRs shall be sent to the originator along with the CCB rationale for rejection. Deferred CRs shall be filed, to be sent back to the board at the proper time.

The CMO, acting as the secretariat of the CCB, shall prepare, distribute, and file the meeting minutes, and shall maintain records of the current status of the CR. These records shall be maintained available for audit by provider and Project QA.

10.2.2.2.5 Implementation

Approved CRs shall result in a change authorization notice which authorizes and directs the implementation of the change in the software and associated documentation.

For the Project, this change authorization will result in technical direction to the provider or in a contract change, depending on the impact of the change. The Software Manager will direct the preparation of the appropriate document.

10.2.2.2.6 Verification

Implemented changes shall be verified by the software provider. If the implementation involves code changes, the provider shall determine if the verification requires the rerun of tests in the test plan or the development of a addition to the test plan. Regression testing shall be included in the tests to assure that errors have not been introduced in existing functions by the change.

After the successful implementation and testing of the change described in the CR, the CMO shall record the occurrence of this process into the change request tracking data base or files.

10.2.2.2.7 Baseline Change Control

Changes to software shall not be recorded as complete until the physical changes have been implemented and tested and the changes to baselined associated documentation have been made and the documents distributed.

The Provider shall document, in a procedure, the initiation, transmittal, review, disposition, implementation, and tracking of change requests and discrepancy reports. In the SMP, the Provider shall use a graphic representation of the change control flow.

10.2.2.3 Change Documentation

The CR form to be used by the Project is shown in Figure FIG4. The documentation of CRs submitted to the Project will include the CR, the analysis report, and the disposition of the CR. Status of the CR will be maintained by the CMO, and monthly reports will be issued that show, for each CR, its number, its title, the submittor, its status, and its disposition.

The provider shall maintain at least the same documentation as the Project, as described in 10.2.2.2. The provider CR form

shall contain at least the same information as the Project CR form. The provider shall describe in the SMP each report used in the configuration management process and explain its purpose and use. This description(s) shall include an example of each report form and cite the location where the information is stored.

10.2.2.4 Change Review Process

The Project review process begins with the screening of each CR by the CMO. Each CR is checked for completeness and clarity, before being accepted from the originator. After a complete and clear CR is obtained, the CMO sends it to the appropriate technical manger for review and analysis. Each change is assessed in terms of impact to system functionality, utility, cost, and schedule. Each change will also be analyzed for impact on software safety, reliability, maintainability, transportability, and efficiency.

The analysis report, which describes the changes that would have to be made to implement the CR, the CSCIs and documents that would have to be changed, and the resources needed to do the change, is sent back to the CMO.

The CMO prepares a review package for each CR, containing the change proposal, relevant documents, and the analysis by the developers, and sends it to the CCB members. The CMO, who is the CCB Secretary, prepares the meeting agendas and records the meeting minutes.

At a CCB meeting, each CR on the agenda is covered in turn. Each member discusses the pros and cons of accepting the CR from their point of view and within their areas of expertise. The CCB chairperson, who is the software manager, is responsible for making the final decision.

The disposition of the CR will be recorded by the CMO, who will maintain files of CRs, the analysis reports, the disposition, and the implementation process.

The provider shall describe in the SMP the process by which each control and review board for configuration management carries out its responsibilities and functions and how each board will provide historical traceability with respect to the configuration identification scheme.

10.2.3 Configuration Status Accounting

Configuration status accounting is the process that provides for traceability of changes to the software. It ensures that status is recorded, monitored, and reported on both pending and completed actions affecting software baselines. The process also defines the current as-built status of the code and associated documentation.

The Project CMO and librarian will be responsible for configuration status accounting and the record keeping and reporting activities that constitute it. Records shall be kept that contain the identifications of the initial software and associated documents and their current status, status of evolving baselines, status of proposed and approved changes, and the implementation status of approved changes. Reports shall be issued that document the information contained in the records.

Each provider shall describe in the SMP the configuration status accounting process to be used and the records and reports to be issued. Records and reports shall, as a minimum, contain the information the Project's records contain as defined in this section.

10.2.4 Configuration Authentication

Configuration authentication is the verification that a deliverable software baseline contains all of the items which are required, and that these items have themselves been verified, i.e., they satisfy their requirements. The authentication function usually consists of two "audits": a functional configuration audit and a physical configuration audit. Functional audits authenticate that the software has been tested to assure that it performs in accordance with requirements in the baseline documentation. Physical audits authenticate that the software to be delivered contains all of the required components, documents, and data.

The Provider shall describe the approach to Functional Configuration Audits (FCAs), and Physical Configuration Audits (PCAs) in the SMP. As a minimum, an FCA and a PCA shall be conducted before each delivery of code products to the Project. The Project shall be notified of all FCAs and PCAs to be conducted by the provider and provision shall be made for Project participation in each.

11.0 DELIVERY AND OPERATIONAL TRANSITION PLAN

TBS

12.0 ABBREVIATIONS AND ACRONYMS

CASE	Computer Aided Software Engineering
CCB	Configuration Control Board
CDR	Critical Design Review
CM	Configuration Management
CMO	Configuration Management Officer
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CR	Change Request
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSU	Computer Software Unit
DID	Data Item Description
FCA	Functional Configuration Audit
GFS	Government Furnished Software
GMI	GSFC Management Instruction
HOL	High Order Language
I/O	Input/Output
ICD	Interface Control Document
ITSG	Independent Software Test Group
NPV	Numerical Progress Value
NRCA	Nonconformance Reporting and Corrective Action
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
PMS	Project Measurement System
SAM	Software Assurance Manager
SCM	Software Configuration Management
SDF	Software Development Folder
SDL	Software Development Library
SEE	Software Support Environment
SM	Software Manager
SMP	Software Management Plan
SMR	Software Management Review
SPAR	Standard Payload Assurance Requirements
SQA	Software Quality Assurance
SQE	Software Quality Engineering
SRMB	Software Risk Management Board
SRR	Software Requirements Review
SSR	Software Specifications Review
TRR	Test Readiness Review
V&V	Verification and Validation
WBS	Work Breakdown Structure

13.0 GLOSSARY

TBS